Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Εργαστηριακή Άσκηση Εαρινό Εξάμηνο 2020-21

Μιχαήλ Μαναγούδης 1059398

Γενικές Πληροφορίες:

Η εργασία υλοποιήθηκε σε Python 3 (έκδοση 3.9.5 στο περιβάλλον Visual Studio Code) την οποία μπορεί κάποιος να κατεβάσει και να εγκαταστήσει από τον σύνδεσμο.
Χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

- numpy
- pandas
- scipy
- matplotlib
- scikit-learn
- tensorflow

Η εγκατάστασή τους μπορεί να γίνει από την γραμμή εντολών/cmd (Command Prompt) με την εντολή **pip install lib** , όπου lib κάθε μία από τις παραπάνω βιβλιοθήκες.
Επιπλέον στα αρχεία κώδικα υπάρχουν σχόλια για την καλύτερη κατανόηση του κώδικα
Λόγω περιορισμένου όγκου αρχείου που μπορεί να κατατεθεί στο eclass χρειάζεται να κατεβεί το αρχείο glove.6B.200d.txt και να τοποθετηθεί στον φάκελο code.

Ερώτημα 1 :

Αρχικά εισάγονται οι βιβλιοθήκες που θα χρειαστούν και έπειτα φορτώνονται τα δεδομένα από το αρχείο *healthcare-dataset-stroke-data.csv* μέσω της βιβλιοθήκης pandas.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_score, recall_score, f1_score

df = pd.read_csv('healthcare-dataset-stroke-data.csv') # read data
```

A. Στην συνέχεια τυπώνεται μία γενική περιγραφή των δεδομένων (πόσα παραδείγματα υπάρχουν, τα γνωρίσματά τους καθώς και ο τύπος των γνωρισμάτων τους κ.α.)

```python
print(df.info(),'\n')
```

```
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

Έπειτα για κάθε γνώρισμα (στήλη) των δεδομένων τυπώνεται μία περιγραφή με τις βασικές πληροφορίες. Συγκεκριμένα οι στήλες χωρίζονται σε 3 κατηγορίες:

1. Στήλες που περιέχουν strings
2. Στήλες που περιέχουν μόνο τιμές 0 ή 1
3. Στήλες που περιέχουν αριθμούς

Και αντίστοιχα για κάθε μία από αυτές τυπώνονται τα βασικά χαρακτηριστικά:

1.

```
Column: smoking_status
count             5110
unique               4
top       never smoked
freq              1892
Name: smoking_status, dtype: object
Unique values:  ['formerly smoked' 'never smoked' 'smokes' 'Unknown']
```

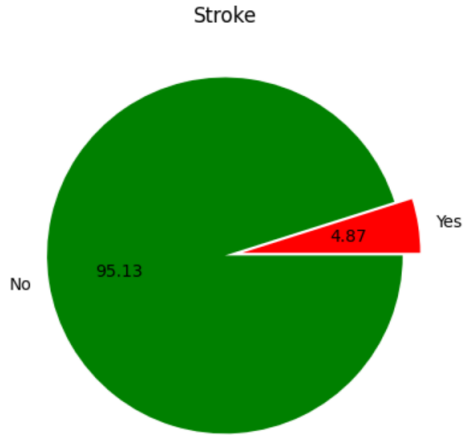
2.

```
Column: stroke
Count:  5110
value | times
0    4861
1     249
Name: stroke, dtype: int64
```

3.

```
Column: avg_glucose_level
count    5110.000000
mean      106.147677
min        55.120000
max       271.740000
Name: avg_glucose_level, dtype: float64
```

Τέλος εμφανίζεται ένα διάγραμμα πίτας για το γνώρισμα ενδιαφέροντος (κλάση), το αν έχει πάθει κάποιος εγκεφαλικό ή όχι.



B. Για τον χειρισμό των ελλιπών τιμών αρχικά γίνεται μία προεπεξεργασία στα δεδομένα ώστε όσες στήλες έχουν τιμές 0 ή 1 (δυαδικές τιμές) να αντικατασταθούν από No ή Yes αντίστοιχα. Επίσης στην στήλη smoking_status (όπως διευκρινίζεται στην εκφώνηση) οι τιμές Unknown αντικαθίστανται με Nan (αντικείμενο της numpy) για να αντιμετωπιστούν αργότερα ως ελλιπείς τιμές και όχι σαν αποδεκτές. Θεωρούμε ότι στην στήλη gender η τιμή Other θεωρείται αποδεκτή και όχι σαν ελλιπής εφόσον δεν διευκρινίζεται στην εκφώνηση, σε αντίθετη περίπτωση θα μπορούσε να αντιμετωπιστεί όπως την τιμή Unknown.
Παρατηρούμαι από τα δεδομένα ότι ελλιπείς τιμές έχουν μόνο οι στήλες smoking_status και η bmi.
Οπότε για τις ζητούμενες μεθόδους χειρισμού των ελλιπών τιμών αφού δημιουργηθεί ένα αντίγραφο των δεδομένων (ώστε τα αρχικά να παραμείνουν χωρίς τις επακόλουθες αλλαγές):

1. Για κάθε στήλη αναζητείται αν αυτή περιέχει ελλιπείς τιμές και αν ναι διαγράφεται ολόκληρη η στήλη.
2. Για κάθε στήλη αναζητείται αν αυτή περιέχει ελλιπείς τιμές και αν ναι τότε στην περίπτωση που οι τιμές είναι strings αντικαθίσταται με την

επικρατέστερη/συχνότερο εμφανιζόμενη τιμή της στήλης αυτής,
ενώ στην περίπτωση που είναι αριθμοί αντικαθίσταται με τον μέσο όρων αυτών.

3. Για την αντικατάσταση των ελλιπών τιμών με Linear Regression χρησιμοποιείται η μέθοδος interpolate των DataFrames (πίνακας των δεδομένων) της pandas με όρισμα μεθόδου linear. Ουσιαστικά η μέθοδος αυτή αντικαθιστά τις nan τιμές χρησιμοποιώντας γραμμική παρεμβολή.
4.

Γ. Για την υλοποίηση του κατηγοριοποιητή με την μέθοδο Random Forest χρησιμοποιείται η αντίστοιχη συνάρτηση *RandomForestClassifier()* αφού προηγουμένως χωριστούν τα δεδομένα σε δεδομένα εκπαίδευσης και δεδομένα αξιολόγησης με την συνάρτηση *train_test_split()* της scikit-learn με ποσοστό 75%-25%. Έπειτα από πειραματισμούς παρατηρήθηκε ότι με χρήση 100 estimators έχω καλύτερη ακρίβεια, κάτι το οποίο είναι σχετικό καθώς λόγο της τυχαιότητας που χωρίζονται τα δεδομένα με την *train_test_split()* δεν είναι σταθερή η ακρίβεια κάθε φορά. Τέλος για τον υπολογισμό της απόδοσης του μοντέλου με τις μετρικές F1 score, Precision και Recall χρησιμοποιήθηκε στον υπολογισμό τους η μέθοδος averaging με τιμή *weighted* η οποία υπολογίζει για κάθε κατηγορία (ύπαρξη ή όχι εγκεφαλικού) το σκορ και έπειτα το αθροίζει με βάση τα κατάλληλα βάρη(ανάλογα τις σωστές προβλέψεις σε κάθε κατηγορία). Αυτή η μέθοδος είναι χρήσιμη στην περίπτωσή αυτή που για την θετική τιμή ύπαρξης εγκεφαλικού έχει πολύ λίγα παραδείγματα σε σχέση με αυτήν μη ύπαρξης.

Ερώτημα 2 :

Όμοια με το ερώτημα 1 εισάγονται οι απαραίτητες βιβλιοθήκες και φορτώνονται τα δεδομένα.

Αρχικά διαγράφεται το παράδειγμα με nan τιμή στην θέση 1466 και αναδιατάσσεται η αρίθμηση.

Για να χρησιμοποιηθούν τα δεδομένα στο νευρωνικό δίκτυο χρειάζεται να μετατραπούν τα κείμενα email σε διανύσματα με βάση την τεχνική Word Embeddings. Ουσιαστικά κάθε λέξη που περιέχεται στα δεδομένα σε διάνυσμα 200 διαστάσεων.

Στη συνέχεια υλοποιείται το μοντέλο νευρωνικού δικτύου μέσω της keras. Στην συγκεκριμένη περίπτωση χρησιμοποιείται ένα Recurrent Neural Network (RNN) το οποίο περιλαμβάνει έναν ειδικό τύπο layer, τον Long Short-Term Memory (LSTM), ο οποίος επιτρέπει την αποθήκευση δεδομένων από προηγούμενες εισόδους του δικτύου. Αυτό είναι πολύ χρήσιμο στην συγκεκριμένη περίπτωση που η είσοδος του δικτύου είναι χρονικά

μεταβαλλόμενη και η σειρά με την οποία εισάγονται τα δεδομένα έχει σημασία. Παρακάτω φαίνεται η δομή του νευρωνικού δικτύου.

```
Model: "Spam_mails"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
LSTM (LSTM)                  (None, 60)                62640
_________________________________________________________________
Dense1 (Dense)               (None, 32)                1952
_________________________________________________________________
Dropout1 (Dropout)           (None, 32)                0
_________________________________________________________________
Dense2 (Dense)               (None, 16)                528
_________________________________________________________________
Dropout2 (Dropout)           (None, 16)                0
_________________________________________________________________
Output (Dense)               (None, 1)                 17
=================================================================
Total params: 65,137
Trainable params: 65,137
Non-trainable params: 0
_________________________________________________________________
```

Η είσοδος έχει διάσταση 200 αφού όπως εξηγείται παρακάτω χρησιμοποιούνται διανύσματα 200 διαστάσεων από το μοντέλο GloVe για τις λέξεις. Έπειτα από πειραματισμούς βρέθηκαν τα πλήθη των νευρώνων κάθε layer. Το δίκτυο αποτελείται από 2 Dense layers και ενδιάμεσα τους υπάρχουν επίπεδα Dropout για να αποφευχθεί το Overfitting. Στα κρυφά επίπεδα χρησιμοποιείται η relu συνάρτηση ενεργοποίησης ενώ στην έξοδο (αφού είναι binary) η sigmoid. Ως loss function ορίστηκε η binary crossentropy και σαν μετρητική η binary accuracy.

Όμοια πάλι με την ερώτηση 1 τα δεδομένα χωρίζονται τυχαία σε δεδομένα εκπαίδευσης και αξιολόγησης με την συνάρτηση *train_test_split()* της scikit-learn με ποσοστό 75%-25%. Τέλος η εκπαίδευση έγινε σε 60 epochs με batch size 32 (διαρκεί περίπου 2-3 λεπτά).

Τα αποτελέσματα φαίνονται παρακάτω (οι μετρικές είναι οι ίδιες με αυτές του ερωτήματος 1).

```
f1_score: 97.04%,   precision: 97.10%,   recall: 97.07%
```